A comparative analysis of LSTM and GRU for short term weather forecasting

Er. Sanjay Kumar Yadav[1]*, Sankalpa KC[2]

[1]Research Supervisor: Dept. of Computer Science, St. Xavier's College, Maitighar, Kathmandu, Nepal
[2]LSRA Research Scholar: Bsc.CSIT 3rd year/5th semester, St. Xavier's College, Maitighar, Kathmandu, Nepal
Correspondence: Er. Sanjay Kumar Yadav; sanjay@sxc.edu.np*

Abstract
Accurate weather forecasting is critical for a wide range of applications, from agriculture and transportation to disaster management. In recent years, deep learning techniques, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have shown promise of improving the accuracy of various sequence-to-sequence processing tasks, such as Natural Language Processing and prediction models. In this study, a comparative analysis has been conducted for LSTM and GRU with encoder-decoder models using hourly MERRA-2 data to predict the temperature and humidity for prediction windows of 5 hours, 10 hours, and 20 hours. The results indicate that for a prediction window of 5 and 10 hours, both the models performed similarly, albeit LSTM performed slightly better. However, for the 20- hour window, LSTM performed significantly better. The results also indicate that errors in forecasted values increased with the increment of output prediction hour window.

Keywords: Weather forecasting, Deep learning, RNN, LSTM, GRU, Time series, MERRA-2.

1. Introduction

In many industries, agriculture, aviation, transportation, and disaster management, weather forecasting is essential. Although weather prediction has been done since ancient times, there have been changes in the approach, fashion, and method.

The most substantial advancement in computational weather forecasting was the invention of Numerical Weather Prediction (NWP). NWP is based on computational models for simulating the climate by using complex differential equations relating to climate dynamics. However, it comes with its own set of hurdles (Wiston et al., 2018).

With the advancement in internet technology, large amounts of labeled datasets have emerged. Similarly, computation power has also increased since the advent of

GPUs. This, coupled with the rapidly increasing cloud computing industry, has made the idea of using machine learning very fascinating and realistic. Weather data is a time series data, data/observations that are observed over a period of time. Weather data being a time series demonstrates temporal dependence, which denotes that there is some relationship between past and future observations.

This relationship between past and future data/observations is the basis for the prediction of time series. Machine learning is an excellent way to realize hidden patterns and relationships between data.

The weather conditions of a particular area may be different from the much wider regional weather, as there are many uncertainties and unseen variables that affect it. In this research, weather data from one specific location is used to train the machine learning models such that weather conditions for that location could be predicted. This research focuses on weather forecasting by implementing deep learning algorithms, using a multivariate, multi-step time series approach.

The hourly temperature and humidity are predicted using past observations. Temperature (Celsius), humidity (g/kg), precipitation corrected (mm/hr), and wind speed (m/s) at 10 meters are the variables used as the input. Two types of Recurrent Neural Networks are used – Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). The models are trained for 5, 10, and 20-hour output windows with the same 24-hour input window. Python and the following libraries – Pandas, Numpy, Scikit-learn, Matplotlib, and Tensorflow 2 with sequential API – were used for the implementation.

2. Related works

Karna et al. (2021) uses a linear regression model to predict the maximum temperature in Pokhara. Using the maximum temperature as input, the model gives its predicted maximum temperature value as the output. This is a univariate approach to weather forecasting, as only one input variable is taken into account for both input and output.

Ji et al. (2012) uses a rule-based model for hourly rainfall prediction. It uses 13 input variables to predict and estimate the hourly rainfall. In this paper, Classification And Regression Trees (CART) and C4.5 were used to make decision trees. In this paper, multiple weather input variables were taken to estimate/predict rainfall. This paper focuses on the transparency of the prediction models rather than black box approaches like neural networks.

Researchers have used the Box-Jenkins and Holt-Winters methods for forecasting meteorological time series. Murat et al. (2018) uses ARIMA and its seasonal variant (SARIMA) for daily mean temperature and daily precipitation prediction. This approach used the models separately for temperature and precipitation forecasting.

In 2009, Yalavarthi and Mogalla published a paper that, based on the daily maximum temperatures for a period of n days prior at a particular location, the maximum temperature of the next day at that site could be forecast. The researchers used the Support Vector Machine (SVM) and found that it consistently outperformed the Multilayer Perceptron (MLP) which was trained using the back-propagation approach when the results were compared.

According to a review paper published in *Nature* by LeCun et al. (2015), the main advantage of deep learning is to learn complicated and nonlinear correlations between variables and outcomes.

A multivariate time-series analysis method presented in a study uses neural networks. In their studies, the price of flour was modeled and predicted using feedforward networks. The model could recognize the price curve and anticipate prices with high accuracy. These findings indicate that the neural network technique is the statistical modeling approaches' direct competitor. (Chakraborty et al., 1992).

In a study, ARIMA and LSTM, two representative forecasting methods for time series data, were evaluated for accuracy. The results showed that LSTM was superior to ARIMA when these two approaches were used for a financial dataset. In particular, the LSTM-based algorithm outperformed ARIMA by 85% on average. The research also claims that changing the number of epochs has no positive effect on the prediction accuracy of LSTM. (Siami-Namini et al., 2018).

A paper by Sutskever et al. (2014) demonstrates the use of a multilayered LSTM to convert the input sequence of English language words into French words. This research uses the encoder-decoder architecture. It converts the input sequence of variable lengths to a fixed size vector and converts the vector to an output sequence.

A study compared regular RNN (i.e, just the tanh gate), LSTM, and GRU for the task of sequence modeling. Polyphonic music modeling and speech signal modeling were used for this research. It was found that both LSTM and GRU performed better than the traditional tanh or vanilla RNNs. However, there was no concrete decision between the performance of LSTM and GRU (Chung et al., 2014)

3. Methodology

3.1 Data collection

Weather data was collected from the National Aeronautics and Space Administration (NASA) Langley Research Center (LaRC) Prediction of Worldwide Energy Resource (POWER) Project, funded through the NASA Earth Science/Applied Science Program. The meteorological data is obtained from NASA's GMAO MERRA-2 assimilation model.

According to Luo et al. (2020), Huang et al. (2022), and Huang et al. (2021), the accuracy of the weather data from the MERRA-2 dataset, when compared with the observed in-situ measurements, was found to be relatively accurate, and a conclusion that it was fit for research purposes was drawn.

Native resolution hourly data was taken starting from 01/02/2005 until 11/15/2022. Latitude 27.6938 and Longitude 85.324 (Maitighar area) were taken as the area for weather data collection consisting of four weather variables, totaling 156,630 hourly observations:

Table 1: Data information

T2M: MERRA-2 Temperature at 2 Meters (C)
QV2M: MERRA-2 Specific Humidity at 2 Meters (g/kg)
PRECTOTCORR: MERRA-2 Precipitation Corrected (mm/hour)
WS10: MERRA-2 Wind Speed at 10 Meters (m/s)

3.2 Data preprocessing

The dataset presented a native resolution data collected from satellites. Removal of outliers was not done. Some preliminary preprocessing was done to the dataset – setting the date-time index and converting the object datatype of the data to numeric using Pandas library.

The date-time index in the dataset was encoded using cyclical encoding. The date-time was encoded as sin value of hour and day, and cos value of hour and day, resulting in 4 new variables in our dataset. This provides information about the periodic and cyclical nature of the data to the model for better understanding the relationship between them. The previous 156,630 rows × 4 columns of data after the encoding operation transformed to 156,630 rows × 8 columns. The reason for choosing this encoding is that it preserves the relationship between the temporal data and reduced dimensionality as opposed to other methods like one-hot encoding.

3.3 Sliding window

A sliding window protocol/methodology was used to transform the time series problem to a supervised learning problem. Sequences of training examples were created, such that a 24 hour/time-step window taken as input had a label of 5, 10 and 20-hour window. The window "slides" across the data, so that each segment overlaps the previous one. As such, all data points are covered using the sliding window method.

For illustration purposes, an example of sliding window consisting of 24 timestamps as the input window and 3 timestamps as the output is created as:

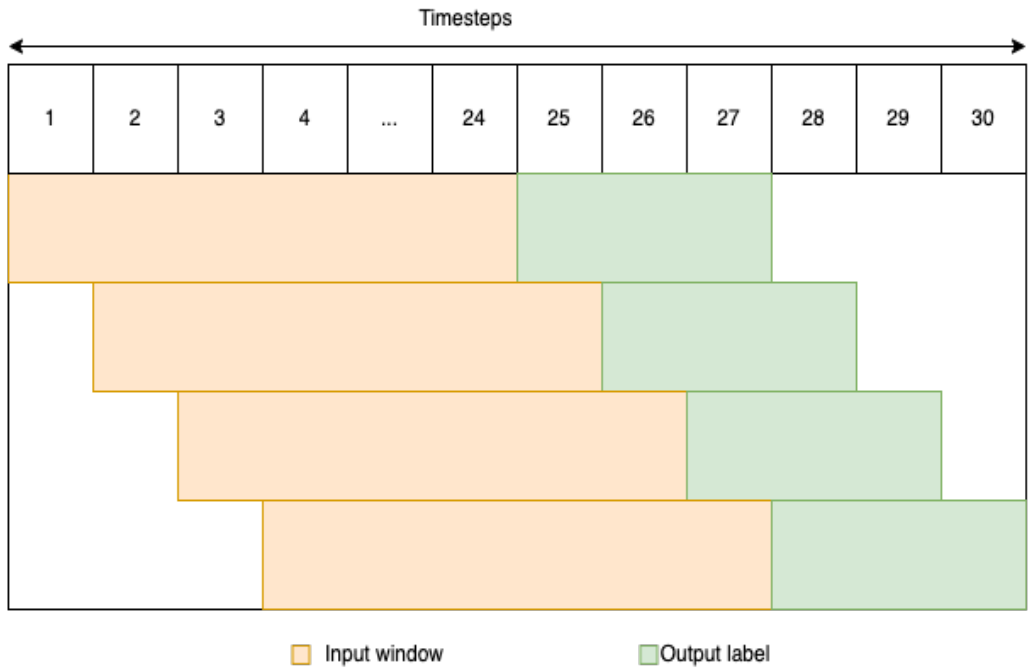

Figure 1: Sliding window model with 24 hours as input and 3 hours as output.

3.4 Data splitting

The data was split into three parts, a training set, validation set, and test set. From the 156,630 rows of data, the training set was aggregated as 0 to 135,000, validation set from 135,000 to 150,000, and the remaining 6,630 as the test set.

3.5 Model architecture

Two types of Recurrent Neural Networks (RNN) are used – LSTM and GRU. Recurrent Neural Networks are a type of neural networks, which can capture and maintain information from recent events. However, vanilla RNN has a fault called vanishing/exploding gradient, which makes it ineffective to learn long-term dependencies. A solution to this problem was proposed by Horchreiter and Schmidhuber in 1997, a new variant called LSTM.

LSTM uses three gates: input, forget, and output. LSTM could store the long-term dependency from farther sequences as well as short-term dependency. Again in 2014, another simpler alternative to LSTM, with only two gates, reset, and update was created by Cho et al. This was the birth of GRU, which seemed effective for sequence modeling.

These models were used in accordance with an auto-encoder. The model architecture starts with an input layer, then proceeds to an LSTM/GRU layer, which encodes the data to fixed size vectors. Then the vector is duplicated according to the size of the output window. Again the duplicated vectors are passed to another layer of LSTM/GRU and finally, a time-distributed layer consisting of two dense layers that specify the temperature and pressure as output variables. The models were created for 3 time windows – 5-hour output, 10-hour output, and 20-hour output.

For ensuring fairness, all models were given similar conditions. Every model consisted of two layers of LSTM/GRU. It was specified that the first layer should have 64 individual LSTM/GRU cells, followed by a repeat vector, and the second layer of 32 LSTM/GRU cells. Both these layers used the ‘relu’ function as their activation function. The learning rate was specified as 0.0001 and the optimizer used was Adam optimizer. The model loss was calculated as the Mean Absolute Percentage Error (MAPE) and batch size was specified as 32.

To prevent overfitting of the model, some strategies were implemented. The models were trained for a total of 200 epochs. The validation dataset was provided, and the model weights were updated only if there was improvement in the validation loss. Early stopping was ensued with a patience value of 20 epochs, such that if there was no improvement in the validation loss for 20 epochs, the last best weight was saved and the training halted.

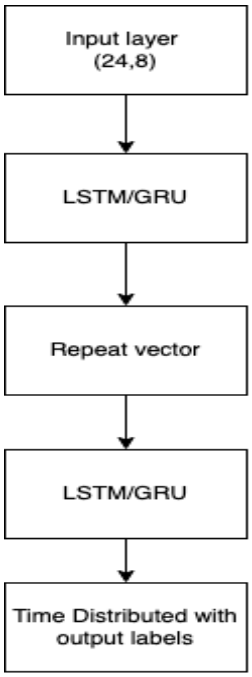

Figure 2: Model architecture

3.6 Accuracy assessment

The trained model is then fit with the test data. Its value is compared with the actual data, and the accuracy is determined using accuracy metrics. The accuracy is determined for both temperature and humidity values separately. The metrics used for the accuracy assessment were:

Mean absolute error (MAE): (1/n) * ∑|actual - predicted|
Mean Squared Error (MSE): (1/n) * ∑(actual - predicted)^2

Mean Absolute Percentage Error (MAPE): (1/n) * ∑(|(actual - predicted) / actual| * 100)

The MAE indicates the absolute error of the actual versus predicted data. The MSE is useful for identifying large errors/deviations. And MAPE gives an average percentage difference between the actual and predicted data or relative error.

3.7 Model deployment

The dataset was obtained from the source, preprocessed, partitioned into windows of input dimension (24,8) and varying output. The dataset was then split into training, validation, and test sets. The training dataset was fed into the model along with the validation set. After the training of the models, the test dataset was then used to forecast the weather variables, and its accuracy was measured by comparing the predicted values with the real values.

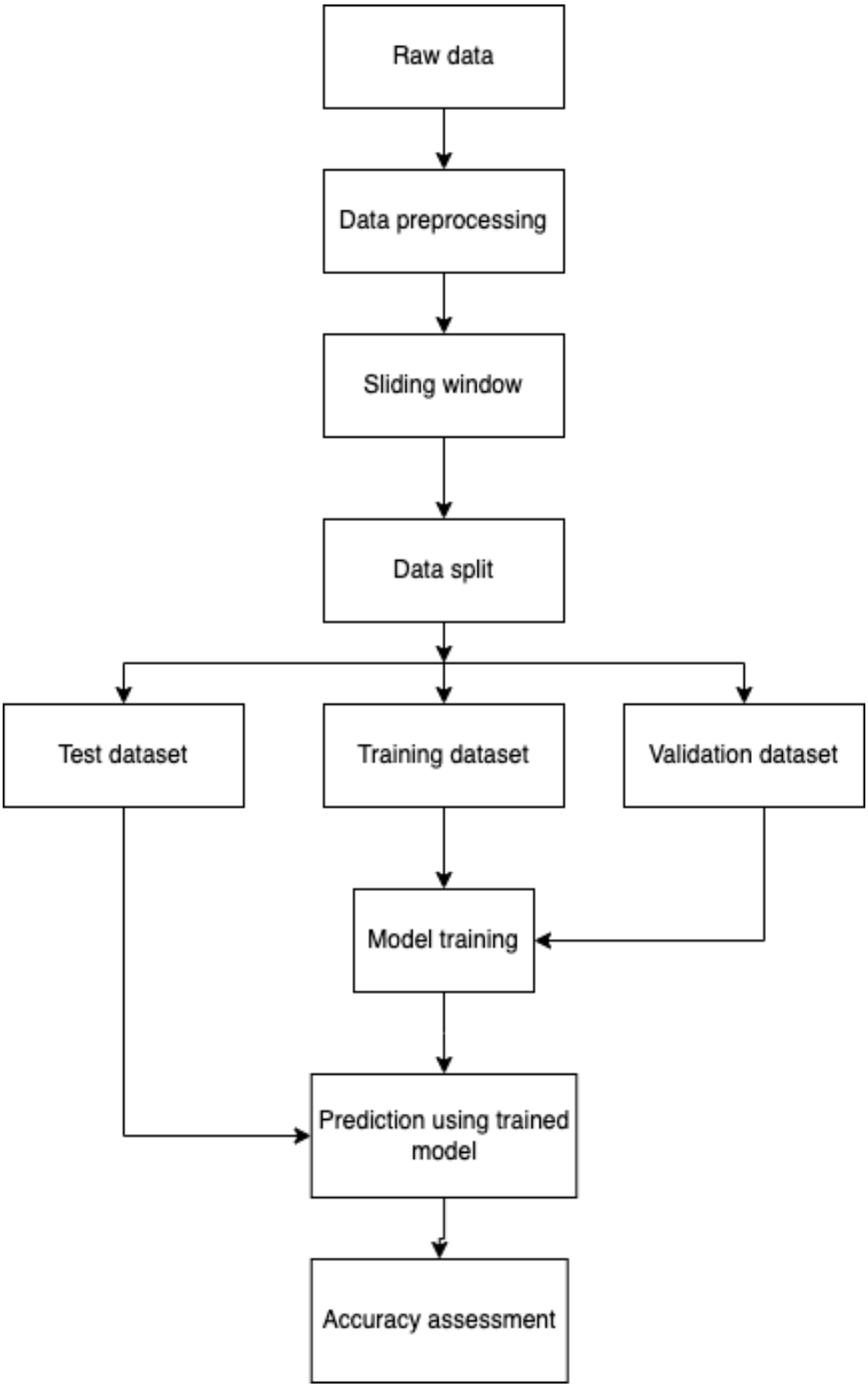

Figure 3: Methodology flowchart

4. Result

4.1 Training result

The training results were obtained as:

Table 2**:** Final model training and validation loss (MAPE).

model/ window	Window 5		Window 10		Window 20	
	Train loss	Validation loss	Train loss	Validation loss	Train loss	Validation loss
LSTM	2.5251	2.8986	3.6692	4.2289	4.8546	5.9933
GRU	2.5278	2.8398	3.6829	3.9516	5.197	5.5866

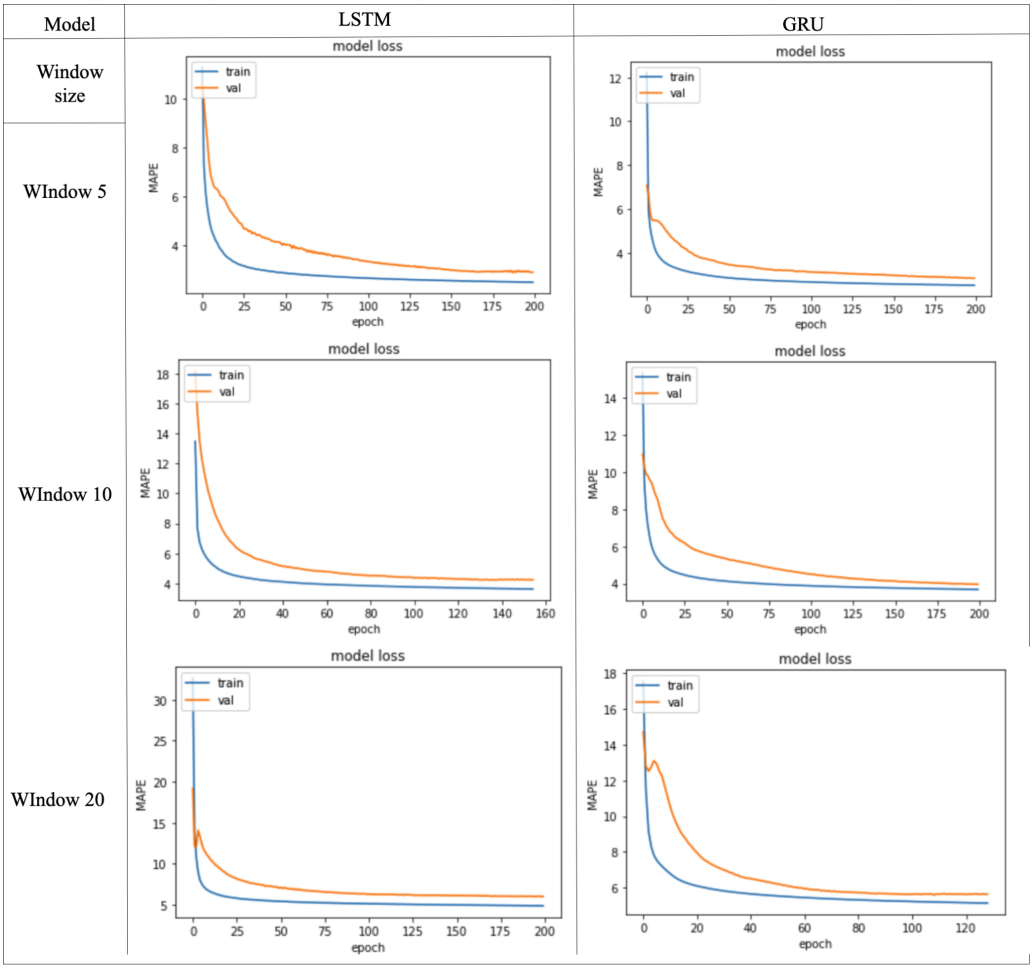

Figure 4: Graph of training and validation loss (MAPE).

4.2 Test result

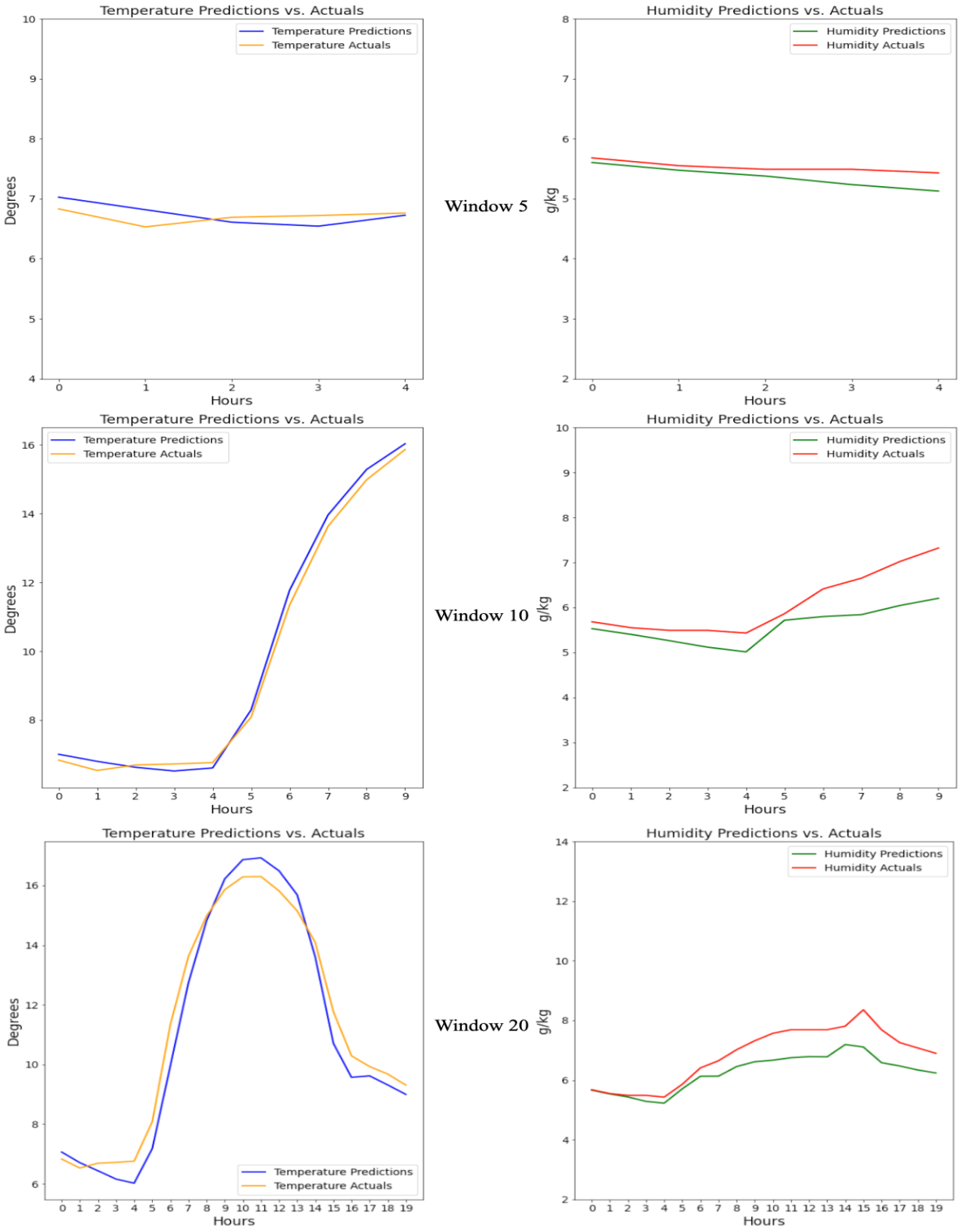

Figure 5**:** Comparison of actual versus predicted temperature and humidity of LSTM

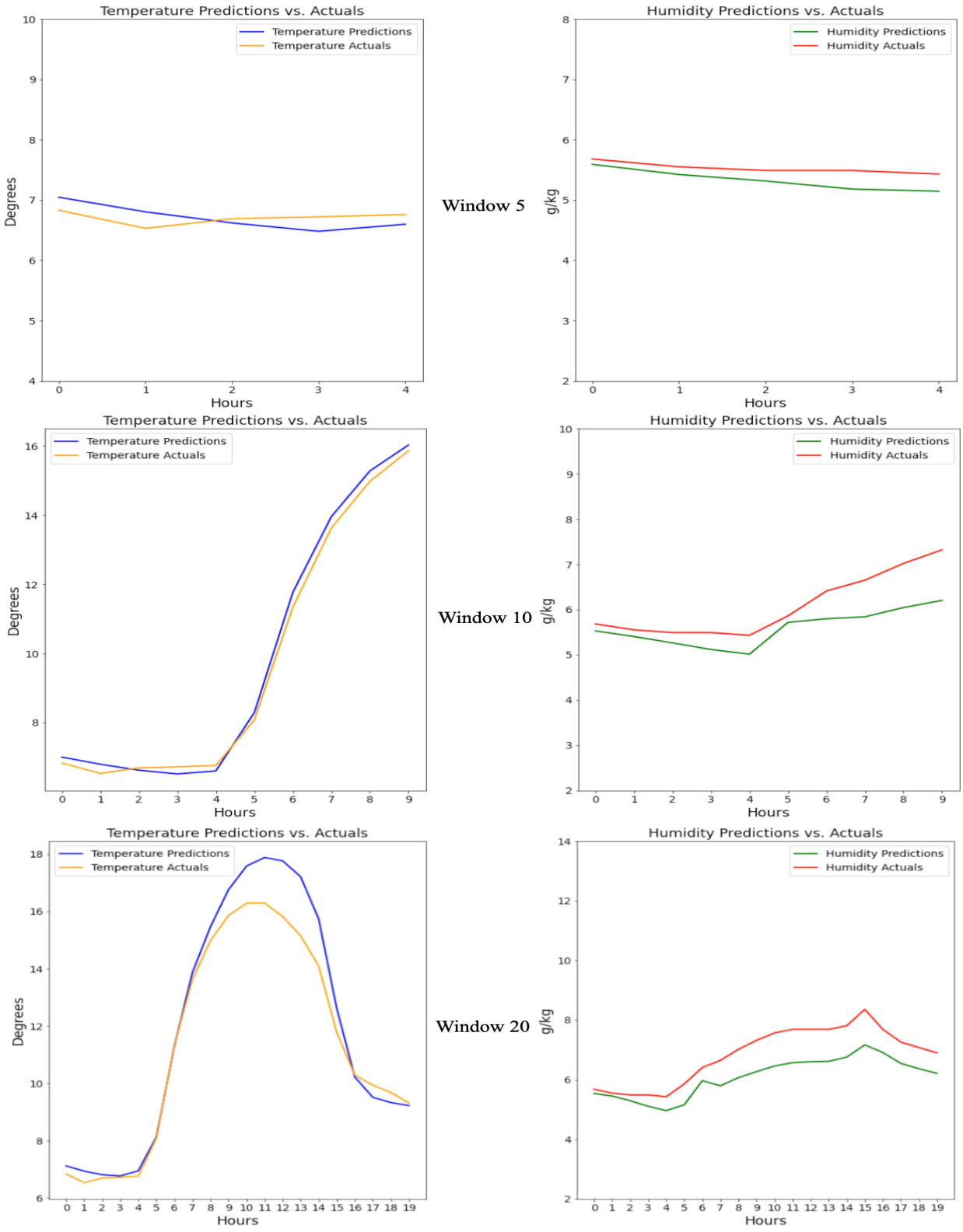

Figure 6**:** Comparison of actual versus predicted temperature and humidity of GRU

The accuracy metrics of test data were obtained as:

Table 3: Window 5 accuracy metrics

Window 5			
	Error metric	Temperature	Humidity
	MAE	0.16	0.16
LSTM	MSE	0.03	0.04
	MAPE	2.33	3
	MAE	0.19	0.2
GRU	MSE	0.04	0.05
	MAPE	2.87	3.59

Table 4: Window 10 accuracy metrics

Window 10			
	Error metric	Temperature	Humidity
	MAE	0.29	0.37
LSTM	MSE	0.16	0.24
	MAPE	2.75	5.55
	MAE	0.23	0.5
GRU	MSE	0.06	0.37
	MAPE	2.49	7.75

Table 5: Window 20 accuracy metrics

Window 20

	Error metric	Temperature	Humidity
	MAE	0.57	0.58
LSTM	MSE	0.42	0.47
	MAPE	5.37	7.84
	MAE	0.65	0.74
GRU	MSE	0.87	0.66
	MAPE	4.95	10.4

4. Discussion

By analyzing Table 3, it was found that for both LSTM and GRU, the accuracy was high. There were no large errors in both the models for both temperature and humidity. LSTM's absolute and relative accuracy was slightly better for both temperature and humidity.

From Table 4, it was found that the values of temperature and humidity were different. When looking at the temperature accuracy metrics, MSE values indicated that GRU had less large deviations as compared to LSTM. Also the absolute and relative errors for temperature were better for GRU. For humidity, LSTM outperformed GRU in all metrics by a small difference. The MSE values indicate that LSTM had fewer significant deviations as compared to GRU.

From Table 5, it was found that the values of temperature and humidity were different. When looking at the temperature accuracy metrics, the MSE values indicated that LSTM had significantly fewer significant deviations as compared to GRU. Also the absolute error for temperature was better for LSTM. However, relative error was less for GRU as compared to LSTM. This indicated that there were relatively larger absolute errors for the data forecasted by GRU, even though the rest of the forecast was relatively accurate. For humidity, LSTM outperformed GRU in all metrics by a significant difference. Both absolute and relative errors were significantly better for LSTM. The MSE values for both LSTM and GRU are high, but LSTM still outshone GRU in terms of relatively less large deviated values.

The 5-hour window accuracy suggests that both models are highly accurate, however LSTM slightly outperformed GRU. GRU for the 10-hour window performed very well for the temperature but badly for humidity. In the 20-hour window, it can be seen that LSTM produced a consistent output with less large deviations.

5. Conclusion

By analyzing the results, it can be seen that the size of the output windows was directly proportional to the error. The results indicated that for the window size of hours 5 and 10, the significant difference between the forecasted value from LSTM and GRU was little; however, both the models produced accurate enough results to be useful. However, for the window size of 24, LSTM performed much better than GRU, indicating that LSTM could produce a much more reliable forecast as compared to GRU.

6. Limitations and recommendations

The major limitations of this research were that the choices of output variables were limited to two – temperature and humidity – and the simplicity of the model. The future prospects of this research include increasing the number of output variables and changing the model complexity by introducing varying numbers of LSTM/GRU cells and/or increasing the number of layers.

7. Acknowledgements

This paper is based on the ‘Fr. Locke and Stiller Research Award’ cycle V. The completion of this research would not have been possible without cooperation from Fr. Dr. Augustine Thomas, S.J. (Principal of St. Xavier’s College, Maitighar) and Mr. Niraj Nakarmi (Head of Research of Science and Technology of St. Xavier’ College, Maitighar). The authors are grateful to Mr. Jeetendra Manandhar, Head of ICT, Mr. Ganesh Yogi, Head of the Department of Computer Science, Er. Rajan Karmacharya, CTO, and Er. Sarjan Shrestha (Research Coordinator, Department of Computer Science) for their tremendous support and assistance.

8. Conflict of Interest

The authors declare no conflict of interest.

9. References

Chakraborty, K., Mehrotra, K., Mohan, C. K., & Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks. Neural networks, 5(6), 961-970.

Charlton, M., & Caimo, A. (2012). Time series analysis (Doctoral dissertation, ESPON| Inspire Policy Making with Territorial Evidence).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Huang, L., Fang, X., Zhang, T., Wang, H., Cui, L., & Liu, L. (2023). Evaluation of surface temperature and pressure derived from MERRA-2 and ERA5 reanalysis datasets and their applications in hourly GNSS precipitable water vapor retrieval over China. Geodesy and Geodynamics, 14(2), 111-120.

Huang, L., Mo, Z., Liu, L., Zeng, Z., Chen, J., Xiong, S., & He, H. (2021). Evaluation of hourly PWV products derived from ERA5 and MERRA-2 over the Tibetan Plateau using ground-based GNSS observations by two enhanced models. Earth and Space Science, 8(5), e2020EA001516.

Ji, S. Y., Sharma, S., Yu, B., & Jeong, D. H. (2012, August). Designing a rule-based hourly rainfall prediction model. In 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI) (pp. 303-308). IEEE.

Karna, N., Roy, P. C., & Shakya, S. (2018). Temperature prediction using regression model. Adv. Eng. ICT Converg. Proc, 4, 161-170.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Luo, B., Minnett, P. J., Szczodrak, M., Nalli, N. R., & Morris, V. R. (2020). Accuracy assessment of MERRA-2 and ERA-Interim sea surface

temperature, air temperature, and humidity profiles over the atlantic ocean using AEROSE measurements. Journal of climate, 33(16), 6889-6909.

Murat, M., Malinowska, I., Gos, M., & Krzyszczak, J. (2018). Forecasting daily meteorological time series using ARIMA and regression models. International agrophysics, 32(2).

NASA Langley Research Center. (01/02/2005 - 11/15/2022). Prediction of worldwide energy resource (POWER) datasets. Retrieved from https://power.larc.nasa.gov/ [Accessed on [11/25/2022].

Radhika, Y., & Shashi, M. (2009). Atmospheric temperature prediction using support vector machines. International journal of computer theory and engineering, 1(1), 55.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1394-1401). IEEE.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

Wiston, M., & Mphale, K. M. (2018). Weather forecasting: From the early weather wizards to modern-day weather predictions. Journal of Climatology & Weather Forecasting, 6(2), 1-9. [Original source: https://studycrumb.com/alphabetizer]